Môn: phân tích thiết kế thuật toán

PHÂN TÍCH BÀI TOÁN NUMBER PARTITIONING

3rd February 2021

Giảng viên: Phạm Nguyễn Trường An

Võ Huy Khôi - 18520949
Hứa Văn Sơn - 18521344
Nguyễn Thịnh Quyền - 18521322

Contents

1 Giới thiệu bài toán

Partition hay number partitioning là bài toán xem xét 1 tập hợp số nguyên dương có thể chia thành 2 tập hợp con có tổng bằng nhau hay không.

1.1 Mô tả hình thức

- Input: list các số nguyên arr, n số phần tử trong arr
- Output: True nếu arr có 2 tập hợp S1, S2 với sum(S1) = sum(S2). False nếu arr không có.
  Điều kiện: n * (sum/2) $< 10^8$

1.2 Ứng dụng bài toán

Bài toán number partitioning có rất nhiều ứng dụng như:

- Ứng dụng trong phân chia công việc.
- Thao túng các cuộc bầu cử.

1.3 Một ứng dụng thực tế

Khi phân chia lượng công việc cho nhóm với một số lượng công việc nhất định, áp dụng number partioning để tìm ra lượng công việc bằng nhau để chia cho 2 người, nếu không có lượng công việc bằng nhau có thể nâng cấp bài toán lên thành chia công việc sao cho tỉ lệ chênh lệch công việc là thấp nhất.

2 Phương pháp thiết kế thuật toán

Dynamic programming

Dynamic programming được phát minh bởi nhà toán học Richard Bellman vào năm 1953 là phương pháp giúp tối ưu thời gian chạy cho những bài toán có dạng bài toán con gối nhau (overlapping subprolem) và cấu trúc con tối ưu (optimal substructure).

Giải quyết bài toán bằng cách chia bài toán thành các bài toán con nhỏ hơn và lưu lại kết quả cho những lần tính toán tiếp theo. Từ đó giải quyết cho toàn bộ bài toán.

Ý tưởng: Áp dụng phương pháp của bài toán subset sum cho number partitioning. Nếu trong list ban đầu có subset có tổng bằng sum/2 với sum là tổng tất cả phần

tử trong list. Tạo một mảng 2 chiều với hàng là subset trong list theo thứ tự và cột là tổng từ 1 đến sum/2.

Ưu điểm: tiết kiệm thời gian tính toán.

Nhược điểm: tốn nhiều bộ nhớ với việc các phần tử trong list có giá trị càng lớn hoặc càng nhiều phần tử thì mảng 2 chiều càng lớn.

2.1 Mã giả

Parttion(arr, n):

1. sum = 0
2. for i = 0 to n: sum+=1
3. if sum div 2 !=0 : return False
4. Initalize M[i][j] = True for i = 0 to n+1, for j = 0 to sum/2 + 1
5. M[i][0] = False for i = 1 to sum/2+1
6. for i = 1 to sum/2 +1
    7. for j = 1 to n+1
    8. M[i][j] = M[i][j-1]
    9. if i>= arr[j-1]: M[i][j] = M[i][j] | M[i-arr[j]][j-1]
10. return M[sum/2][n]

2.2 Tính độ phức tạp bằng lý thuyết

- Tại dòng 1, 3: phép gán và phép so sánh => độ phức tạp là 3
- Tại dòng 2: vòng lặp chạy từ 0->n và thực hiện phép cộng mỗi vòng lặp=> độ phức tạp thời gian n
- Tại dòng 4: Khởi tạo gán M trong 2 vòng lặp n và sum/2 => độ phức tạp n*sum/2
- Tại dòng 5: vòng lặp chạy từ 1-> sum/2 + 1 và thực hiện phép gán => độ phức tạp sum/2
- Tại dòng 6,7,8,9: phép gán dòng 8 có độ phức tạp 1, phép so sánh dòng 9 có độ phức tạp 2 vì trong điều kiện có phép gán và phép logic. Dòng 8, 9 lặp lại sum/2*n lần => độ phức tạp n*sum/2

Áp dụng quy tắc cộng: độ phức tạp của thuật toán là n*sum/2 + n*sum/2 + n + sum/2 + 3.
Áp dụng qui tắc bỏ hằng số và lấy Max: độ phức tạp thời gian là O(n*sum).
Độ phức tạp không gian: O((n+1)*(sum/2 + 1)) = O(n*sum)

2.3 Mã nguồn cài đặt

2.4 Phân tích input, output kiểm tra độ đúng đắn

- Mảng rỗng hoặc chỉ có 1 phần tử: return False
- Mảng có tổng lẻ: return False
- Mảng có n phần tử bằng nhau: nếu n lẻ return False, n chẵn return True

2.5 Phân tích độ phức tạp thuật toán bằng thực nghiệm

size	sum//2	n	operations	297176.98605673*sqrt(n)	MSE	5.22949542e+08*log(n)	MSE	17.00518159*n	MSE	0.86625649*nlog(n)	MSE	4.70586713e-08*n^2	MSE
13	13196	171548	2929817	123085814.4	123050940.1	9093178828	9093178356	2917204.891	271557.13	2583973.799	1380908.057	1384.876287	2929816.673
14	14976	209664	3579591	136074612.8	136027522.2	9244555390	9244554697	3565374.393	318711.728	3210676.952	1582727.153	2068.651797	3579590.402
15	24947	374205	6386781	181789990.5	181677763.3	9681610296	9681608190	6363423.977	545716.806	6001280.11	2185316.586	6589.596661	6386777.601
16	28295	452720	7724906	199953962.2	199804686.7	9825311709	9825308673	7698585.809	637141.463	7368222.513	2320230.53	9644.928724	7724899.979
18	43487	782766	13350924	262924556.8	262585367.9	10238422002	10238413297	13311077.97	1030715.7	13275524.67	1416903.836	28833.91194	13350892.86
26	102572	2666872	45439987	485306669.3	483174679.5	11163255958	11163163476	45350642.64	2848092.51	49315083.03	19162071.8	334690.9768	45438754.39
33	126079	4160607	70857143	606168719.6	602013107.8	11498804146	11498585829	70751877.56	3860898.33	79249418.95	35492755.46	814616.217	70852460.19
36	161439	5811804	98962918	716424870	709556858.4	11750966246	11750549522	98830782.39	5112298.13	113128297.9	54811975.3	1589503.834	98950152.18
42	169753	7129626	121374338	793502954.5	784165294.3	11905152872	11904534144	121240584.8	5696536.02	140601009.7	70971219.52	2392065.998	121350764.1
46	243107	11182922	190353812	993785879.6	975384950	12244755728	12243276043	190167619.3	8417262.4	226825635.3	123350294.2	5885051.29	190262818
50	296047	14802350	251937116	1143353454	1115251008	12456304000	12453755941	251716649.7	10537489.2	305426406.2	172664354.1	10311005.02	251726028.8
60	383752	23025120	391812131	1425988834	1371104449	12789622856	12783619834	391546346.7	14429288.3	487805120.4	290580607.6	24948444.05	391017034.3
86	623764	53643704	912568643	2176579591	1976035725	13427725982	13396680317	912220927.7	25189428.1	1193185431	768706670.1	135418243.3	902465194.7
94	714746	67186124	1142880941	2435874827	2151118158	13597556171	13549441205	1142512239	29028075.3	1513307426	991928686.4	212421677.9	1122966552
124	918252	113863248	1936596215	3171078243	2511042079	13995555784	13860922654	1936265209	35804212.8	2639735616	1793822517	610108108.5	1837980684
163	1371220	223508860	3801025445	4442859153	2300261512	14504401306	13997494876	3800808751	40586571.4	5370086300	3793419623	2350872889	2986836302
196	1888969	370237924	6295938008	5718151069	2634688550	14885171670	13488124416	6295963129	17785463.7	9128963365	6610532256	6450620091	1404159593
150	71810	10771500	183190629	975333781.7	957975562.9	12216475645	12215102062	183171313.5	2660162.65	217976077.8	118130283.7	5459992.326	183109243.5
181	181404	32834124	558365513	1702855870	1608709442	13057366375	13045422362	558350241	4129711.62	710178856.8	438841614.4	50732998.18	556055940.5
218	333182	72633676	1235110489	2532702541	2211127368	13656375075	13600407430	1235148850	9734548.97	1643085545	1083619947	248265121.1	1209901793
					1164737751		11951707166		10931194.1		818746048.1		570352264.6

trong đó: sum: tổng các phần tử trong list
size: số phần tử trong list
n: sum*size
MSE:

- $\sqrt{sum * size} = 1164737751$
- $\log(sum * size) = 11951707166$
- sum*size = 10931194.11
- sum*size * $\log(sum * size) = 818746048.1$
- $(sum * size)^2 = 570352264.6$

Có thể thấy MSE của sum*size là bé nhất, như vậy độ phức tạp của thuật toán theo thực nghiệm là O(sum*size), bằng với độ phức tạp phân tích lý thuyết.

2.6 Thuật toán tối ưu hóa không gian

Thay vì tạo một mảng 2 chiều có kích thước bằng (size+1)*(sum/2 +1) Tạo một mảng với số phần tử bằng sum/2. + 1 Phần tử thứ j sẽ là True có tập hợp con có tổng bằng j, ngược lại là False. Do đó độ phức tạp không gian là O(sum).

Điều kiện: size*sum $< 10^{10}$

Mã giả: Partition(arr, sum):

1. n = len(arr)
2. sum = 0
3. For i=0 to n : sum+=arr[i]
4. If sum div 2 != 0 : return False
5. Initalize Part = [False] * ((sum//2)+1)
6. For i to n:
    7. For j = Sum//2 to arr[i] -1 :
        8. Part[j] = True If Part[j - arr[i]] == True | j == arr[i]
9. Return Part[Sum//2]

Phân tích độ phức tạp bằng phân tích lý thuyết

- Tại dòng 1,2: phép gán => độ phức tạp 2
- Dòng 3: vòng lặp từ 0 đến n và phép gán trong mỗi vòng lặp => độ phức tạp n
- Dòng 4: phép chia và so sánh => độ phức tạp là 2
- Dòng 5: khởi tạo mảng Part với kích thước sum//2 + 1 và gán cho các phần tử của Part => độ phức tạp là sum
- Dòng 6,7,8 khởi tạo 2 vòng lặp từ 0 đến n và từ sum//2 đến arr[i] - 1 => độ phức tạp là sum*n

Độ phức tạp của thuật toán: sum*n + sum + n + 2 + 2

Áp dụng quy tắc lấy Max: độ phức tạp: O(sum*n)

Thực nghiệm:

size	sum//2	n	operations	1314348.55111567*sqrt(n)	MSE	2.78020017e+10*log(n)	MSE	5.98030359*n	MSE	0.24164799*nlog(n)	MSE	3.80289889e-09*n^2	MSE
150	69233	10384950	60047581	4235578700	4235153033	4.49165E+11	4.49165E+11	62105153.77	15853647.5	40543188.31	44294038.72	410131.9458	60046180.36
165	128267	21164055	122811378	6046582621	6045335289	4.68959E+11	4.68959E+11	126567474.1	30605406.9	86266244.54	87411495.93	1703383.914	122799564.5
239	423398	101192122	600824889	13221596597	13207937993	5.12461E+11	5.12461E+11	605159610.5	72302191.6	450728348.9	397283907	38941097.29	599561621.7
288	659023	189798624	1123147015	18107441071	18072574884	5.29947E+11	5.29947E+11	1135053392	163972517	874244144.4	705093180.6	136993795.4	1114760924
316	799135	252526660	1504179155	20886427751	20832194058	5.37886E+11	5.37886E+11	1510186091	134562626	1180605136	932054957.5	242509774.6	1484501243
381	1152715	439184415	2616566178	27544444033	27419882905	5.53272E+11	5.53272E+11	2626456134	227713105	2111993450	1544636601	733514357.9	2511647915
460	1538724	707813040	4239254185	34967922456	34710003239	5.66541E+11	5.66541E+11	4232936864	231347248	3485432763	2413096454	1905249680	3786990850
506	1886090	954361540	5721888601	40603831362	40198645651	5.7485E+11	5.7485E+11	5707371744	407329032	4768417335	3162626327	3463702933	4554423252
556	2237959	1.244E+09	7481454735	46363264695	45755657008	5.82225E+11	5.82225E+11	7441322879	773872579	6296874053	4039992835	5888011013	4615570524
611	2537139	1.55E+09	9159222132	51749157830	50932150809	5.88336E+11	5.88336E+11	9270618358	1432834491	7927168224	4588175456	9138727372	612382306.1
893	4839681	4.322E+09	25984811148	86406148775	82406384072	6.16842E+11	6.16842E+11	25845886161	2683389449	23171220395	11760312743	71031530102	66108001469
982	5201181	5.108E+09	30464746413	93932802185	88855334968	6.21486E+11	6.21486E+11	30544757861	2209402361	27589996640	12918701925	99206856595	94413450427
1080	6295492	6.799E+09	40847597689	1.08377E+11	1.00385E+11	6.29439E+11	6.29439E+11	40660869681	3901270785	37197538834	16878072806	1.75801E+11	1.7099E+11
1306	8167217	1.067E+10	63804137357	1.35744E+11	1.19814E+11	6.41958E+11	6.41958E+11	63788222912	1424977758	59515655440	22996841119	4.32663E+11	4.27932E+11
1579	10586829	1.672E+10	99941311718	1.69936E+11	1.3744E+11	6.5445E+11	6.5445E+11	99970360880	2409827068	95089278781	30761905806	1.0627E+12	1.05799E+12
165	126689	20903685	122301148	6009273588	6008028919	4.68615E+11	4.68615E+11	125010382.4	25884839.5	85142428	87797139.79	1661730.088	122289858.3
262	509084	133380008	787783377	15179445957	15158989970	5.2014E+11	5.2014E+11	797652940.7	125090226	603000725.5	506944547.3	67654432.74	784872936.7
316	826492	261171472	1548322833	21240924689	21184418285	5.38822E+11	5.38822E+11	1561884692	205378175	1223145461	949325432.5	259397778.4	1526439120
419	1399541	586407679	3472608522	31828075171	31638068196	5.61309E+11	5.61309E+11	3506895948	489192439	2860941348	1968254188	1307717924	3216968104
556	2265454	1.26E+09	7516898425	46647198961	46037565194	5.82565E+11	5.82565E+11	7532745095	488351039	6377952586	3978125531	6033576984	4483270091
893	4623053	4.128E+09	24693324600	84450206595	80759377871	6.15568E+11	6.15568E+11	24689003584	461932700	22088373661	11039204175	64814987435	59926807994
1080	6230439	6.729E+09	40136358255	1.07816E+11	1.00066E+11	6.2915E+11	6.2915E+11	40240710056	2896116689	36796277232	16030010476	1.72187E+11	1.67444E+11
1188	7035084	8.358E+09	50006356945	1.20158E+11	1.09258E+11	6.35177E+11	6.35177E+11	49981462464	1577702397	46141061220	19277920127	2.65636E+11	2.60886E+11
1436	9249081	1.328E+10	79110751835	1.51474E+11	1.29173E+11	6.48055E+11	6.48055E+11	79428480475	7097355470	74812124921	25722694663	6.70843E+11	6.66162E+11
1736	12194642	2.117E+10	1.26776E+11	1.91236E+11	1.43175E+11	6.61016E+11	6.61016E+11	1.26602E+11	6624495305	1.21629E+11	35754552000	1.70432E+12	1.6996E+12
2308	17928303	4.138E+10	2.4838E+11	2.67361E+11	98941771961	6.79649E+11	6.79649E+11	2.47456E+11	2.1398E+10	2.44437E+11	44080697809	6.51126E+12	6.50652E+12
2791	23254701	6.49E+10	3.87583E+11	3.34847E+11	1.95189E+11	6.92163E+11	6.92163E+11	3.88145E+11	2.0868E+10	3.90469E+11	47380859196	1.60198E+13	1.60151E+13
					60450445484		5.79534E+11		2211888482		10485616375		4.3106E+11

trong đó: sum: tổng các phần tử trong list
size: số phần tử trong list
n: sum*size
MSE:

- $\sqrt{sum * size} = 60450445484$
- $\log(sum * size) = 5.79534E + 11$
- sum*size = 2211888482
- sum*size * $\log(sum * size) = 10485616375$
- $(sum * size)^2 = 4.3106\text{E}{+}11$

Có thể thấy sum*size có MSE bé nhất phù hợp với độ phức tạp lý thuyết của thuật toán.

2.7 Trace back tìm subset

Bài toán nâng cấp từ number partition: Áp dụng bài toán trên để tìm ra 2 subset có tổng bằng nhau. Thực hiện traceback từ ma trận đã tạo của number partition.

- Input: ma trận partition part, tổng các phần tử trong list sum, list arr
- Output: 2 subset có tổng bằng sum/2.

Mã giả:
Traceback(part, sum, arr):

1. tong = 0
2. index = len(part) - 1
3. path = []
4. while tong!= sum:
    5. i = part[index].index(True) - 1
    6. tong += arr[i]
    7. path.append(i)
    8. index -= arr[i]
9. Initalize S1 = [], S2 = []
10. for i = 0 to len(arr):
    11. S1.append(arr[i]) if i in path
    12. else: S2.append(arr[i])
13. return S1, S2

2.8 Tham khảo

1. https://www.geeksforgeeks.org/partition-problem-dp-18
2. https://en.wikipedia.org/wiki/Partition_problem